



18 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

12 Offenlegungsschrift
10 DE 102 14 393 A 1

5 Int. Cl.⁷:
H 03 M 13/29

21 Aktenzeichen: 102 14 393.5
22 Anmeldetag: 26. 3. 2002
43 Offenlegungstag: 6. 2. 2003

DE 102 14 393 A 1

63 Innere Priorität:
101 16 179. 4 27. 03. 2001
71 Anmelder:
Technische Universität Dresden, 01069 Dresden, DE

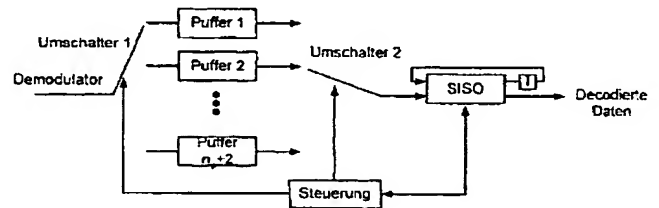
72 Erfinder:
Vogt, Jörg, Dipl.-Ing., 01187 Dresden, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

54 Verfahren zur iterativen Decodierung von verketteten Codes

57 Der Erfindung, die ein Verfahren zur iterativen Decodierung von verketteten Codes betrifft, liegt die Aufgabe zugrunde, den Datendurchsatz der iterativen Decodierschaltung bei Einsatz eines Iterationsabbruchkriteriums zu erhöhen. Dies wird dadurch gelöst, dass die Eingangsdaten in einer geeigneten Art und Weise gepuffert werden. Dadurch kann bei Bedarf eine höhere als die mittlere Iterationszahl durchgeführt werden, obwohl der Datendurchsatz nur für die mittlere Iterationszahl dimensioniert werden muss.



DE 102 14 393 A 1

[0001] Die Erfindung betrifft ein Verfahren zur iterativen Decodierung von verketteten Codes bei dem Zuverlässigkeitsinformationen zwischen den Teildecodern ausgetauscht werden. Bei diesem Verfahren wird das Codewort durch die Kombination von mehreren Unter-codes gebildet.

[0002] Iterative Decodierverfahren sind beispielsweise aus der US 5 446 747, US 5 563 897 und der EP 1085 660 A1 bekannt. In der DE 195 26 416 A1, US 6182261 B1 und DE 196 30 343 A1 wird die Verwendung eines adaptiven Abbruchkriteriums beim iterativen Decodieren übertragener, multidimensional codierter Information beschrieben.

[0003] Iterativ decodierte Codes sind z. B. Turbo-Codes, seriell verkettete Codes und Produktcodes. Zur Decodierung der Teilcodes werden in der Regel Soft-In-Soft-Out Symbol- oder Sequenzschätzer verwendet, die Zuverlässigkeitsinformationen untereinander austauschen.

[0004] Bei einem Verfahren zur iterativen Decodierung von verketteten Codes, bei dem Zuverlässigkeitsinformationen (L-Werte) zwischen verschiedenen Teildecodern ausgetauscht werden, wird das Codewort durch die Kombination mehrerer Unter-codes gebildet. Die Decodierung dieser Codes erfolgt in der Regel iterativ, d. h. es besteht eine Rückkopplung zwischen den Teildecodern. Dabei werden nacheinander die Teilcodes decodiert und das Decodierungsergebnis, die sogenannte extrinsische Information, als a-priori Information den anderen Decodern zur Verfügung gestellt. Beispiele für iterativ decodierte Codes sind Turbo-Codes und seriell verkettete Codes. Turbo-Codes wurden erstmalig in G. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error-correcting coding: Turbo codes" Proc. 1993 International Conference Communication, Genf, Mai 1993, Seiten 1064-1070, vorgestellt. Seriell verkettete iterativ decodierte Codes sind in S. Benedetto, G. Montorsi, "Serial concatenation of block and convolutional codes" Electron. Lett., vol. 32, no. 10, Seiten 887-888, Mai 1996 beschrieben.

[0005] Turbo-Codes werden seit wenigen Jahren eingesetzt und sind als grundlegendes Verfahren zur Kanalcodierung für den Mobilfunkstandard der dritten Generation "IMT-2000" vorgeschlagen worden. Turbo-Codes haben den Vorteil eines hohen Codiergewinnes und können flexibel an die Kanalerfordernisse angepasst werden.

[0006] Zur iterativen Decodierung werden Decoder für die Teilcodes verwendet, welche neben den Zuverlässigkeitswerten des übertragenen Codeworts außerdem a-priori Informationen nutzen können. Als Decodierungsergebnis wird eine extrinsische Information berechnet, welche den Decodern der anderen Teilcodes wiederum als a-priori Information zur Verfügung steht. Diese Decoder werden allgemein SISO (Soft-In-Soft-Out) genannt.

[0007] Realisierungen eines SISO-Decoders sind z. B. MAP oder SOVA-Decoder. Der MAP-Algorithmus wurde in L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate" IEEE Transactions on Information Theory, Seiten 284-287, März 1974 beschrieben. Der SOVA-Algorithmus wurde in J. Hagenauer et al. "Iterative ("Turbo") decoding of systematic convolutional codes with MAP and SOVA algorithms", ITG Fachtagung "Codierung", München, Okt. 1994, beschrieben.

[0008] Der Decodierungsgewinn steigt mit der Anzahl der Iterationen bis eine Sättigung erreicht wird. Ab einer Anzahl von 8 Iterationen wird typischerweise nur noch ein relativ geringer Zuwachs am Decodierungsgewinn erzielt, so dass für eine Vielzahl an Anwendungen 8 Iterationen die Obergrenze darstellt. Die Tatsache, dass die volle Iterationsanzahl je-

doch nur für eine relative geringe Menge an Codeworten notwendig ist – viele Codeworte werden schon nach wenigen Iterationen korrekt decodiert – führte zu Verfahren, die eine korrekte Information erkennen und somit die Iteration schon vor der maximalen Iterationszahl abbrechen können. Verfahren zum vorzeitigen abbrechen der Iterationen sind z. B. in R. Y. Shao, S. Lin, and M. P. C. Fossorier "Two Simple Stopping Criteria for Turbo Decoding", IEEE Transactions on Communications, Seiten 1117-1120, August 1999, beschrieben. Ein Verfahren zum vorzeitigen Abbrechen der Iteration ist z. B. das Überprüfen einer dem Codewort beigegebenen CRC nach jeder ganzen oder halben Iteration. Wenn die CRC ein korrektes Ergebnis signalisiert, wird die Iteration abgebrochen. Durch das vorzeitige Abbrechen der Iterationen ist die mittlere Iterationszahl geringer als die maximale Iterationszahl.

[0009] Die mittlere Iterationszahl eines Turbo-Decoders ist stark von der Blocklänge und dem SNR abhängig. Typisch beträgt die mittlere Iterationszahl 0.5 bis 5 Iterationen.

[0010] Der Vorteil des Abbrechens des Iterationsprozesses liegt in dem verringerten Stromverbrauch des Decoders, da die Decodierschaltung für die nicht benötigten Iterationen in einem stromsparenden Ruhemodus verharren kann.

[0011] Der Datendurchsatz des Decoders wird durch das Abbrechen des Iterationsprozesses nicht erhöht, da auf den nächsten Empfangsblock gewartet werden muß. Der Datendurchsatz der bekannten Verfahren mit einer SISO-Einheit ist demzufolge von der maximalen Iterationsanzahl abhängig. Bei parallelen Decoderrealisierungen mit je einer SISO-Einheit für eine halbe Iteration entspricht der Datendurchsatz der gesamten Decoderschaltung dem Durchsatz der SISO-Einheit.

[0012] Der Erfindung liegt die Aufgabe zugrunde, ein Verfahren zur iterativen Decodierung von verketteten Codes anzugeben, dessen Datendurchsatz von der maximalen Iterationsanzahl unabhängig ist und gleichzeitig eine erhebliche Steigerung des Durchsatzes bewirkt.

[0013] Gemäß der Erfindung wird diese Aufgabe durch ein Verfahren in Verbindung mit den im Oberbegriff des Anspruchs 1 genannten Merkmalen dadurch gelöst, dass

die Eingangsdaten blockweise in einer bestimmten Anzahl an Eingangspuffern gespeichert werden, wobei ein Eingangspuffer jeweils einen Empfangsblock speichert und die Anzahl der Eingangspuffer mindestens dem Verhältnis von der maximalen Iterationsanzahl zu der garantierten Iterationsanzahl entspricht.

- die garantierte Iterationsanzahl unter der maximalen Iterationsanzahl liegt,
- ein Iterationsabbruchkriterium ausgewertet wird, und
- der iterative Decodiervorgang abgebrochen wird, wenn entweder die maximale Iterationsanzahl erreicht wird oder wenn alle Eingangspuffer mit Eingangsdaten gefüllt sind oder dass Iterationsabbruchkriterium erfüllt wird.

[0014] Weiterhin erfindungsgemäß wird die Aufgabe durch ein Verfahren in Verbindung mit den im Oberbegriff des Anspruchs 6 genannten Merkmalen dadurch gelöst, dass

- die Eingangsdaten in einem Ringpuffer gespeichert werden und
- die Größe des Ringpuffers mindestens einer Anzahl an Empfangsblöcken entspricht, die dem Verhältnis von der maximalen Iterationsanzahl zu der garantierten Iterationsanzahl entspricht,
- die garantierte Iterationsanzahl unter der maximalen

Iterationsanzahl liegt,

- ein Iterationsabbruchkriterium ausgewertet wird, und
- der iterative Decodiervorgang abgebrochen wird, wenn die maximale Iterationsanzahl erreicht wird oder wenn der verbleibende freie Platz in dem Ringpuffer einen bestimmten Wert unterschritten hat oder dass Iterationsabbruchkriterium erfüllt wird.

[0015] Vorteilhafte Verfahrensvarianten sind Gegenstand der abhängigen Unteransprüche.

[0016] Jeweils zwei SISO-Decoder Operationen werden normalerweise als eine Iteration bezeichnet. Mit einer bestimmten Datenrate des SISO-Decoders d_{SISO} wird eine bestimmte Kanaldatenrate d_{Kanal} unterstützt. Der Zusammenhang ergibt sich aus der maximalen Iterationsanzahl.

$$d_{\text{Kanal}} = d_{\text{SISO}} / (2 \cdot I_{\text{max}})$$

[0017] Bei dem erfindungsgemäßen Verfahren wird der Decoder nicht mit der maximalen Iterationsanzahl dimensioniert, sondern mit einer Iterationsanzahl, welche nur leicht über der mittleren Iterationsanzahl und unterhalb der maximalen Iterationsanzahl liegt. Diese Iterationsanzahl wird als garantierte Iterationsanzahl bezeichnet, da der Decoder diese Iterationsanzahl auf jeden Fall ausführen kann. Die Kanaldatenrate beträgt bei dem erfindungsgemäßen Verfahren

$$d_{\text{Kanal}} = d_{\text{SISO}} / (2 \cdot I_{\text{gar}}).$$

es gibt sich eine höhere Kanaldatenrate, da $I_{\text{gar}} < I_{\text{max}}$.

[0018] Damit sich die Leistungsfähigkeit der Decodierung nicht verschlechtert, muß jedoch auch manchmal ein Empfangsrahmen mit der maximalen Iterationsanzahl decodiert werden. Dieses Problem wird bei dem erfindungsgemäßen Verfahren mit zusätzlichem Speicherplatz gelöst. Wenn ein Empfangsrahmen mehr Iterationen als die garantierte Iterationsanzahl benötigt, wird der nächste Empfangsrahmen nicht sofort decodiert sondern zunächst zwischengespeichert. Es erfolgt also ein Austausch zwischen Datendurchsatz des SISO-Decoders und der erforderlicher Puffergröße.

[0019] Bei Verwendung von Eingangspuffern mit einer Größe die jeweils einem Empfangsrahmen entspricht, ergibt sich die notwendige Pufferanzahl aus dem Verhältnis zwischen maximaler Iterationsanzahl und garantierter Iterationsanzahl.

[0020] Der notwendige Puffer kann aber auch als Ringpuffer anstatt als auch aus mehreren einzelnen Puffern bestehen. Die Größe entspricht dann der Summe der einzelnen Puffer.

[0021] Die erforderliche Puffergröße ist außerdem abhängig von der Datenübergabe an den Turbo-Decoder. Die obigen Angaben bezogen sich auf einen Burstmodus, wo die Daten in einem kurzen Zeitintervall zum Turbo-Decoder übertragen werden. Werden die Daten anstatt im Burstmodus kontinuierlich zum Turbo-Decoder übertragen, ist die Puffergröße um einem Empfangsrahmen größer zu dimensionieren.

[0022] Bei dem Verfahren muss der Datendurchsatz nur für eine garantierte Anzahl an Iterationen dimensioniert werden, die etwas über der mittleren Iterationsanzahl liegen sollte. Durch die erfindungsgemäße Zwischenspeicherung der empfangenen Daten und den gesteuerten Zugriff auf diese kann bei Bedarf eine höhere oder eine niedere als die mittlere vorgegebene Anzahl an Iterationsschritten durchgeführt werden. Ein Überlauf der Daten wird vermieden.

[0023] Der Iterationsprozess wird abgebrochen, wenn entweder die maximale Iterationsanzahl erreicht ist, das Iterationsabbruchkriterium erfüllt ist oder der Zwischenpuffer mit Daten gefüllt ist.

[0024] Dazu wird mittels einer Steuerung ständig der Zustand der Puffer geprüft und die Rückmeldung von der SISO-Einheit über einen Iterationsabbruch abgefragt.

[0025] Vorteilhaft wird als Iterationsabbruchkriterium die korrekte Decodierung der CRC-Information herangezogen oder ein zweimal aufeinanderfolgendes, identisches Decodierergebnis verwendet.

[0026] Es können auch mehrere SISO-Decoder parallel betrieben werden, wobei dann die Zuordnung der Eingangsdaten zu den Puffern und der Puffer zu den SISO-Decodern flexibel mittels einer Kontrollschaltung gestaltet wird. Die Ausgabe der Teilcodes aus den SISO-Decodern wird vorteilhaft einer Sortierung unterzogen, die der Reihenfolge des Eingangs in die Puffer entspricht.

[0027] Der Vorteil der Erfindung besteht darin, dass der Durchsatz der Decodierschaltung erhöht werden kann, da der Durchsatz nach dem erfindungsgemäßen Verfahren nunmehr von der mittleren Iterationszahl abhängig ist.

[0028] Das Verfahren kann prinzipiell für alle iterativen Decodierverfahren verwendet werden.

[0029] Die Erfindung wird nachfolgend anhand von Ausführungsbeispielen näher erläutert. In den zugehörigen Zeichnungen zeigen:

[0030] Fig. 1 ein Blockschaltbild eines Turbo-Code Encoders

[0031] Fig. 2 ein Blockschaltbild eines Turbo-Code Decoders

[0032] Fig. 3 der zeitliche Ablauf des herkömmlichen Verfahrens mit einem SISO-Decoder

[0033] Fig. 4 der zeitliche Ablauf des erfindungsgemäßen Verfahrens mit einem SISO-Decoder

[0034] Fig. 5 ein Blockschaltbild der herkömmlichen Decodierung mit einer SISO-Einheit

[0035] Fig. 6 ein Blockschaltbild des erfindungsgemäßen Verfahrens mit einer SISO-Einheit

[0036] Fig. 7 in Blockschaltbild des erfindungsgemäßen Verfahrens mit n SISO-Einheiten

[0037] Fig. 8 eine Darstellung der Simulationsergebnisse.

[0038] Im Bereich der Kanalcodierung spielen Turbo-Codes aufgrund des hohen erzielbaren Codierungsgewinns eine wichtige Rolle. Turbo-Codes sind parallel verkettete rekursive systematische Faltungscodes. Die Struktur eines herkömmlichen Turbo-Code Encoders mit zwei Faltungscodes der Einflußlänge von drei Symbolen ist in Fig. 1 dargestellt. Ein Faltungscoder verwendet die Informationssymbole u_k direkt während der anderer Faltungscoder diese Informationssymbole verschachtelt (interleaved) verarbeitet. Die Redundanzsymbole des ersten Faltungscoders werden als u_{p_k1} und die Redundanzsymbole des zweiten Faltungscoders werden als u_{p_k2} bezeichnet. Die Redundanzsymbole können mittels eines optionalen Punktierungsblocks selektiv gelöscht werden, um eine bestimmte Coderate zu erzielen. Die verbleibenden Redundanzsymbole bilden zusammen mit den Informationssymbolen das zu übertragende Codewort.

[0039] Der iterative Turbo-Decoder ist eine Decoderanordnung, die für die Decodierung von Turbo-Codes geeignet ist und dort Anwendung findet. Fig. 2 zeigt eine herkömmliche Turbo-Decoder Struktur. Die vom Demodulator kommenden Daten werden den einzelnen SISO-Faltungsdecodern, die den jeweiligen Encodern entsprechen, zugeführt. Der Decodiervorgang erfolgt iterativ, d. h. nach der Decodierung eines Teilcodes erfolgt die Decodierung des anderen Teilcodes unter Einbeziehung des Decodierergebnisses

(extrinsic Information) des vorherigen Prozesses. Der Turbo-Decoder kann mit einem einzigen SISO-Decoder, der zeitmultiplex betrieben wird, oder parallel (zwei SISO für eine Iterationsstufe) aufgebaut werden. Auch Mischformen sind möglich.

[0040] Zur Decodierung der einzelnen Faltungscodes wird generell ein Soft-Input-Soft-Output Decoder (SISO) verwendet. Dieser kann durch verschiedene Algorithmen realisiert werden (MAP, Max-Log-MAP, SOVA).

[0041] Für die Softwerte werden allgemein logarithmierte Wahrscheinlichkeitsverhältnisse (L-Werte) verwendet. Der Betrag $|L(\hat{u}_k)|$ gibt die Zuverlässigkeit der Entscheidung an, das Vorzeichen $\text{sign}(L(\hat{u}_k))$ stellt die harte Entscheidung dar. Als Decodierergebnis des SISO-Decoders wird die extrinsic Information $L_e(\hat{u}_k)$ für das Informationssymbol u_k bereitgestellt, welche zwischen den Decodern der einzelnen Teilcodes ausgetauscht werden. Die abwechselnde Decodierung der Teilcodes kann auch mit einer einzigen SISO-Einheit realisiert werden. Diese decodiert dann zeitmultiplex die beiden Teilcodes. Da für eine sogenannte Iteration immer beide Teilcodes decodiert werden müssen, ergibt sich für eine Decodierung mit N Iterationen mit einem Turbo-Decoder der nur aus einer einzigen SISO-Einheit besteht, demzufolge 2 N Arbeitsschritte für die SISO-Einheit.

[0042] Während des Decodierprozesses wird ständig auf die Eingangsdaten zugegriffen und diese Zugriffe können wegen des Interleavers auch verwürfelt erfolgen. Deswegen wird bei einem Übertragungssystem mit kontinuierlichem Dateneingang neben dem Datenpuffer für die Decodierung ein weiterer Empfangspuffer benötigt, der die aktuell empfangenen Daten aufnimmt. Diese beiden Puffer werden dann wechselseitig umgeschaltet. In Fig. 5 ist dieses Konzept dargestellt. Die vom Kanal empfangenen Daten werden in Puffer 1 abgelegt während der Decoder die bereits empfangenen Daten aus Puffer 2 decodiert.

[0043] Der zeitliche Ablauf dieses Prozesses ist in Fig. 3 dargestellt. Am Beginn wird der Puffer 1 mit den empfangenen Daten gefüllt. Nach dem Ende dieses Arbeitsschrittes kann mit der Decodierung des ersten Blocks begonnen werden, während der zweite Block empfangen wird und in den Puffer 2 abgelegt wird. In dem skizzierten Ablauf werden maximal 8 Iterationen unterstützt. Durch den Einsatz eines Iterationsabbruchkriteriums wie in der zitierten Literatur beschrieben, kann die Iteration auch vor dem maximalen Ende abgebrochen werden. Der Decoder befindet sich dann bis zum Umschaltzeitpunkt der Puffer in Ruhe.

[0044] Die obigen Ausführungen beziehen sich auf den Stand der Technik. Das erfindungsgemäße Verfahren verwendet neben den bereits beschriebenen Datenpuffern weitere Puffer. In dem hier beschriebenen Ausführungsbeispiel wird ein weiterer Puffer eingesetzt. Dadurch wird eine größere Flexibilität in bezug auf die maximale Iterationszahl erzielt. Die Anzahl an mindestens notwendigen Puffern ergibt sich aus dem Verhältnis aus maximaler Iterationsanzahl zu garantierter Iterationsanzahl zuzüglich eines weiteren Puffers, da im Beispiel kontinuierlich Daten empfangen werden, also $8/4+1 = 3$ Puffer. In Fig. 6 ist die entsprechende Schaltung skizziert. Eine Kontrollschaltung erhält die Information über einen Iterationsabbruch von dem SISO-Decoder und kann dann eine flexible Umschaltung zwischen den Puffern vornehmen (n_p ist die Zahl an zusätzlichen Puffern). Der zeitliche Ablauf des erfindungsgemäßen Verfahrens ist in Fig. 4 dargestellt. Der Decoder ist in dem gewählten Beispiel nur für 4 Iterationen ausgelegt (garantierte Iterationen). Da aber wegen dem zusätzlichen Puffer die Daten länger gültig bleiben, können in dem beschriebenen Beispiel auch maximal 8 Iterationen durchgeführt werden. Dies ist im Beispiel Fig. 4 für Block Nr. 2 der Fall. Für das gezeigte Aus-

führungsbeispiel ist hier die Pufferkapazität erschöpft und die weiteren Blöcke können nur noch mit maximal 4 Iterationen decodiert werden. Erst wenn die Pufferkapazität durch Iterationen kleiner als 4 Iterationen wieder erhöht wird, ist wieder eine höhere Iterationszahl möglich. Das Verfahren ist besonders vorteilhaft, wenn die mittlere Iterationszahl unter der dauerhaft garantierten Iterationszahl (4 in diesem Beispiel) liegt.

[0045] In einem weiteren Ausführungsbeispiel können mehrere Empfangsblöcke durch den Einsatz mehrerer SISO-Decoder parallel decodiert werden. In Fig. 7 ist eine entsprechende Anordnung mit n Decodern dargestellt. Für die n SISO-Decoder werden auch n Puffer benötigt. Ein Puffer wird mit den Empfangsdaten beschrieben und mindestens 1 Puffer wird für die Erhöhung der Iterationszahl verwendet. Somit werden insgesamt mindestens $n+2$ Puffer benötigt. Eine Steuerung verbindet mit einem n-Schalter jeweils einen Puffer mit einem SISO-Decoder. Die SISO-Decoder melden einen eventuellen Iterationsabbruch wie im vorherigen Beispiel der Steuerung. Dem SISO-Decoder, der als erster die Decodierung beendet hat, wird der nächste Eingangspuffer zugewiesen. Als Puffer für die Empfangsdaten wird ein Puffer benutzt, welcher aktuell von keinem SISO-Decoder benutzt wird. Sind alle Puffer in Benutzung, wird durch die Steuerung die Decodierung in dem SISO-Decoder beendet, welcher die höchste Iterationszahl erreicht hat. Da die SISO-Decoder durch die unterschiedliche Iterationszahl die Daten im Extremfall in einer falschen Reihenfolge abliefern, wird am Ausgang ein weiterer Puffer benötigt. Die Steuerung stellt durch einen geordneten Zugriff auf diesen Puffer die richtige Reihenfolge der decodierten Daten wieder her.

[0046] Das erfindungsgemäße Verfahren erzielt einen höheren Datendurchsatz als herkömmliche Verfahren bei einer unwesentlichen Verschlechterung der Decodierleistung. Als Beispiel ist in Fig. 8 das Simulationsergebnis der Bitfehler-rate (BER) mit und ohne dem erfindungsgemäßen Verfahren dargestellt. Für die Simulation der Bitfehler-rate wurde ein Turbo-Code mit dem Generatorpolynom $(g_{\text{feedback}}, g_{\text{parity}}) = (13_{\text{oct}}, 15_{\text{oct}})$ der Teilcodes, die Blocklänge 5114 bits, ein AWGN-Kanal und als SISO-Decoder ein MAP Decoder verwendet. Zur Decodierung wird ein Decoder mit einer SISO-Einheit verwendet.

[0047] Das erfindungsgemäße Verfahren wurde für eine dauerhafte Iterationszahl von 5 Iterationen und maximal 10 Iterationen dimensioniert. Zum Vergleich wurden die BER-Kurven des herkömmlichen Verfahrens mit maximal 5 Iterationen und mit maximal 10 Iterationen dargestellt. Weiterhin ist der Verlauf der mittleren Iterationszahl dargestellt. Es ist zu erkennen, dass das erfindungsgemäße Verfahren bis zu 0,4 dB keinen Vorteil zum herkömmlichen Verfahren hat. Ab 0,5 dB ist das erfindungsgemäße Verfahren jedoch annähernd so leistungsfähig wie das herkömmliche Verfahren mit 10 Iterationen. Das bedeutet, dass der Datendurchsatz des erfindungsgemäßen Verfahrens bei annähernd gleicher Leistungsfähigkeit doppelt so hoch ist, wie bei den herkömmlichen Verfahren.

Patentansprüche

1. Verfahren zur iterativen Decodierung von verketteten Codes, bei dem nacheinander die einzelnen Teilcodes des Empfangsblocks mit einem SISO-Decoder decodiert werden, **dadurch gekennzeichnet**, dass die Eingangsdaten blockweise in einer bestimmten Anzahl an Eingangspuffern gespeichert werden, wobei ein Eingangspuffer jeweils einen Empfangsblock speichert und die Anzahl der Eingangspuffer mindestens dem

Verhältnis von der maximalen Iterationsanzahl zu der garantierten Iterationsanzahl entspricht.

die garantierte Iterationsanzahl unter der maximalen Iterationsanzahl liegt,

ein Iterationsabbruchkriterium ausgewertet wird, und der iterative Decodiervorgang abgebrochen wird, wenn entweder die maximale Iterationsanzahl erreicht wird oder wenn alle Eingangspuffer mit Eingangsdaten gefüllt sind oder dass Iterationsabbruchkriterium erfüllt wird.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass als Iterationsabbruchkriterium die korrekte Decodierung der CRC-Information herangezogen wird.

3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass als Iterationsabbruchkriterium ein zweimal aufeinanderfolgendes, identisches Decodierergebnis verwendet wird.

4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass mehrere SISO-Decoder parallel betrieben werden und die Zuordnung der Eingangsdaten zu den Puffern und der Puffer zu den SISO-Decodern flexibel mittels einer Kontrollschaltung gestaltet wird.

5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, dass die Ausgabe des Decodierergebnisses aus den SISO-Decodern einer Sortierung unterzogen wird, die der Reihenfolge des Eingangs in die Puffer entspricht.

6. Verfahren zur iterativen Decodierung von verketteten Codes, bei dem nacheinander die einzelnen Teiles des Empfangsblocks mit einem SISO-Decoder decodiert werden, dadurch gekennzeichnet, dass die Eingangsdaten in einem Ringpuffer gespeichert werden und

die Größe des Ringpuffers mindestens einer Anzahl an Empfangsblöcken entspricht, die dem Verhältnis von der maximalen Iterationsanzahl zu der garantierten Iterationsanzahl entspricht.

die garantierte Iterationsanzahl unter der maximalen Iterationsanzahl liegt,

ein Iterationsabbruchkriterium ausgewertet wird, und der iterative Decodiervorgang abgebrochen wird, wenn die maximale Iterationsanzahl erreicht wird oder wenn der verbleibende freie Platz in dem Ringpuffer einen bestimmten Wert unterschritten hat oder dass Iterationsabbruchkriterium erfüllt wird.

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, dass als Iterationsabbruchkriterium die korrekte Decodierung der CRC-Information herangezogen wird.

8. Verfahren nach Anspruch 6, dadurch gekennzeichnet, dass als Iterationsabbruchkriterium ein zweimal aufeinanderfolgendes, identisches Decodierergebnis verwendet wird.

9. Verfahren nach einem der Ansprüche 6 bis 8, dadurch gekennzeichnet, dass mehrere SISO-Decoder parallel betrieben werden und die Zuordnung der Eingangsdaten zu den Puffern und der Puffer zu den SISO-Decodern flexibel mittels einer Kontrollschaltung gestaltet wird.

10. Verfahren nach Anspruch 9, dadurch gekennzeichnet, dass die Ausgabe des Decodierergebnisses aus den SISO-Decodern einer Sortierung unterzogen wird, die der Reihenfolge des Eingangs in die Puffer entspricht.

Hierzu 5 Seite(n) Zeichnungen

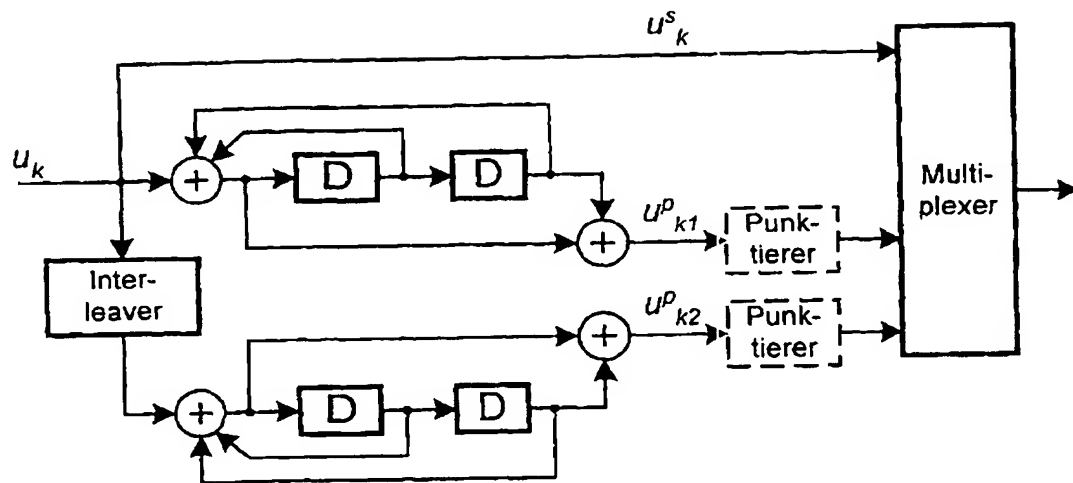


Fig. 1

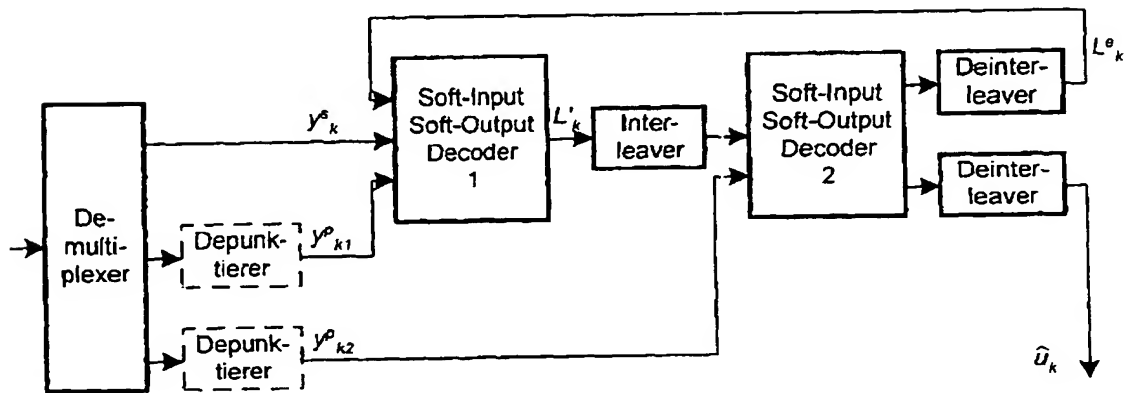
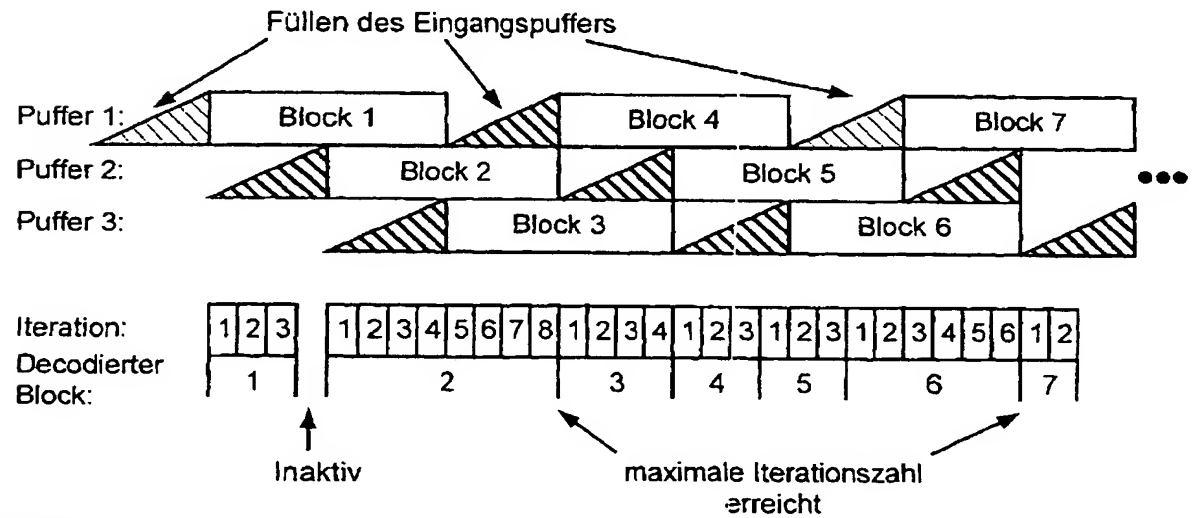
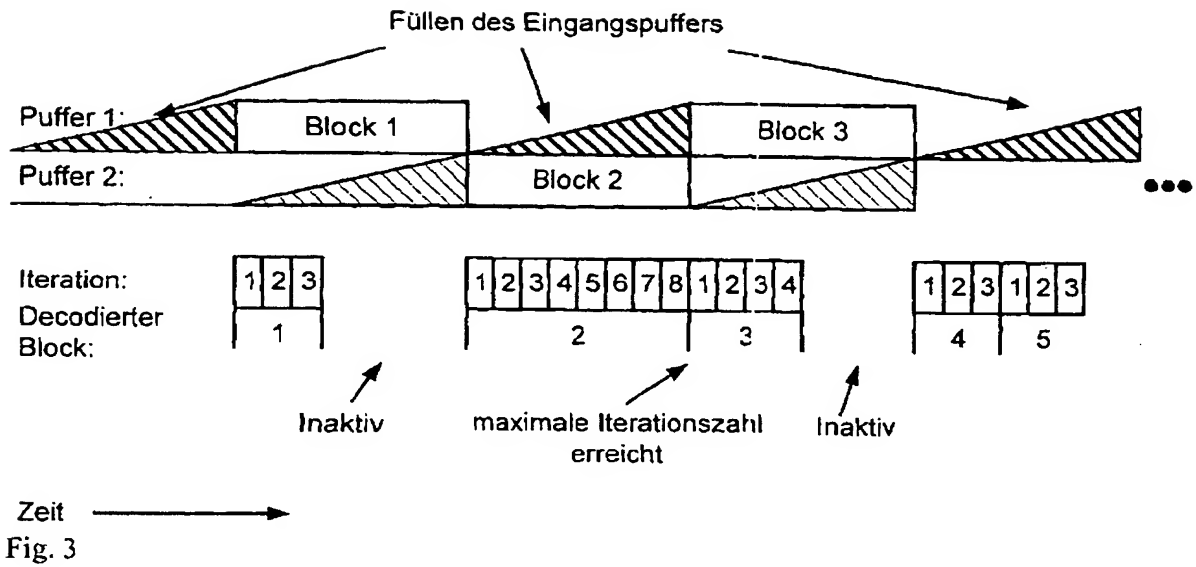


Fig. 2



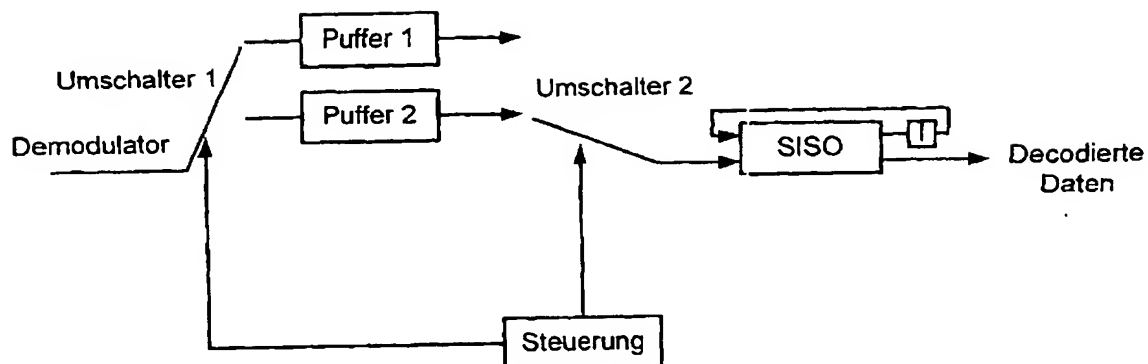


Fig. 5

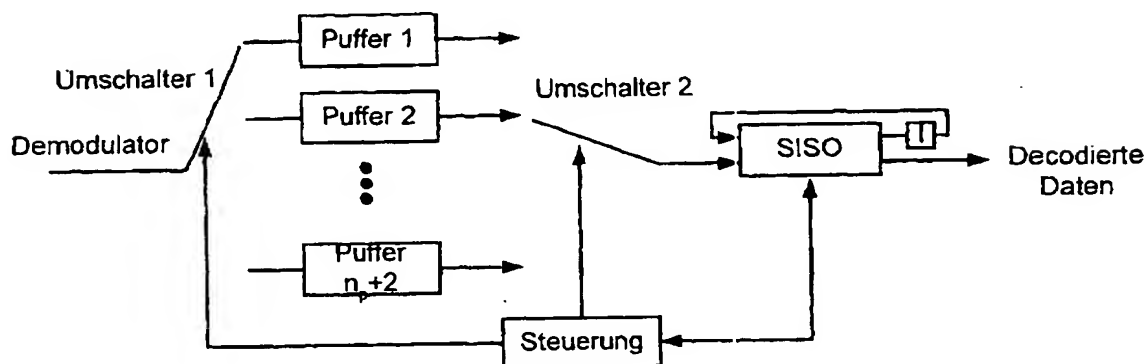


Fig. 6

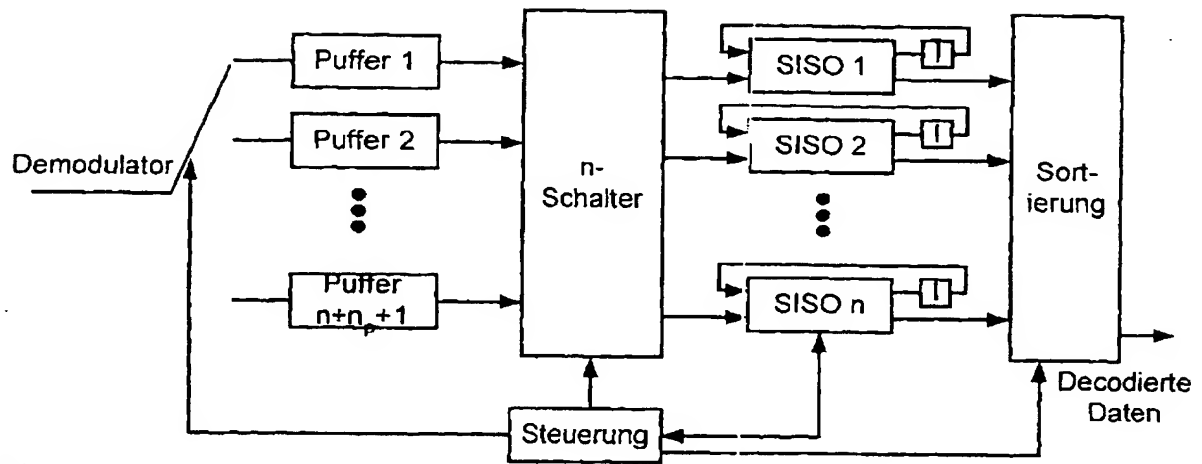


Fig. 7

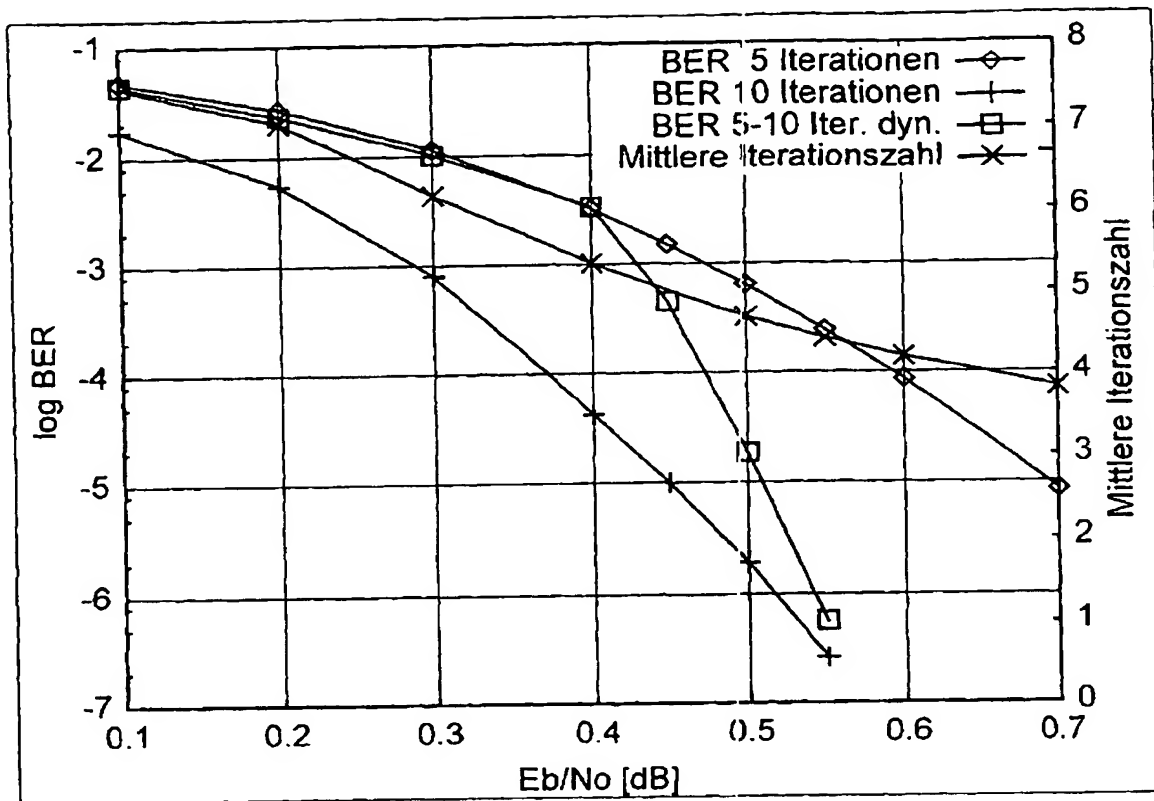


Fig. 8